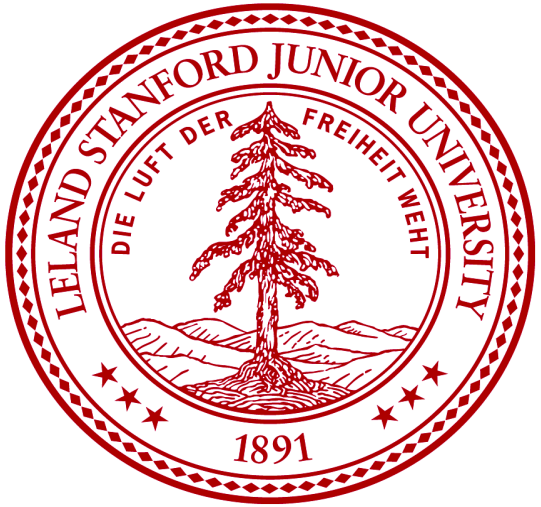
Tree-based Direct Sampling Code Tutorial

Chen Zuo and Zhen Yin

Department of Geological Sciences
Stanford Center for Earth Resource Forecasting



Content

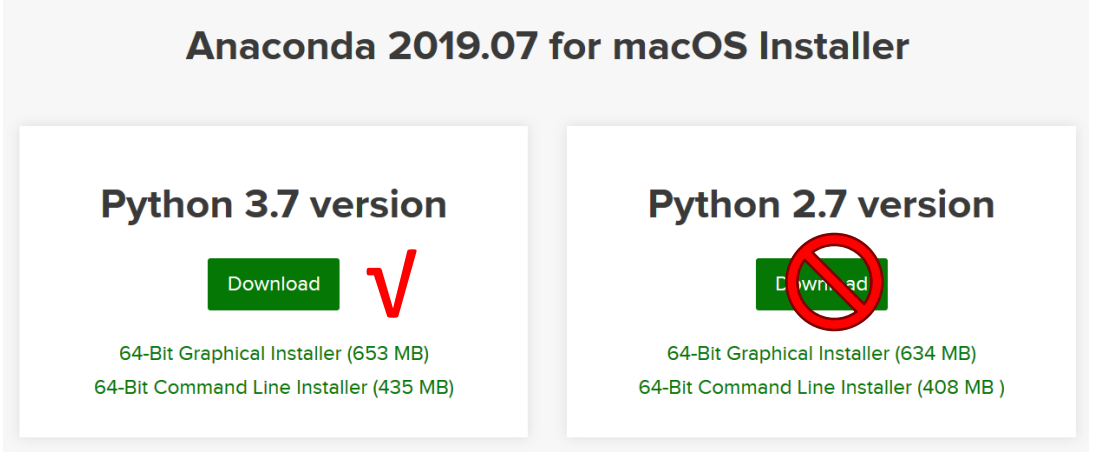
1. Installation
2. Basic steps
3. Example 1: Channel simulation
4. Example 2: Antarctica topography simulation

Installation

In order to perform Tree Direct Sampling (TDS), the following software is necessary:

1. Anaconda and Jupyter: Python 3.7 Version

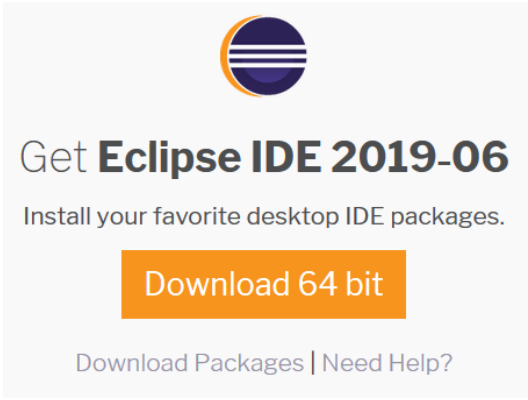
https://www.anaconda.com/distribution/

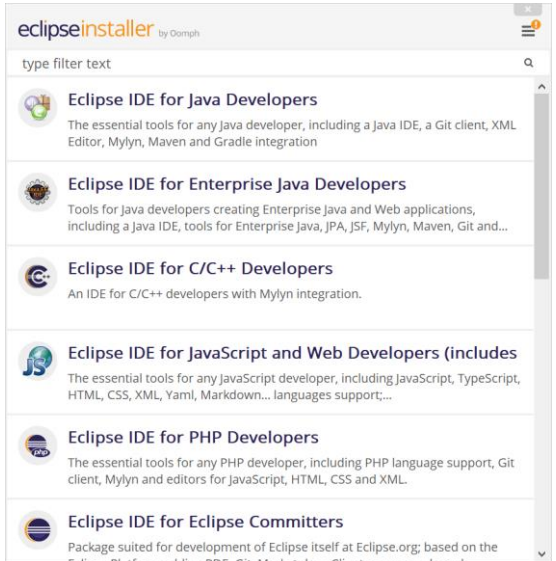


2. Eclipse IDE for Java developer
   Oracle Java SE environment

https://www.eclipse.org/downloads/

https://www.oracle.com/technetwork/java/javase/downloads/index.html





Basic Steps

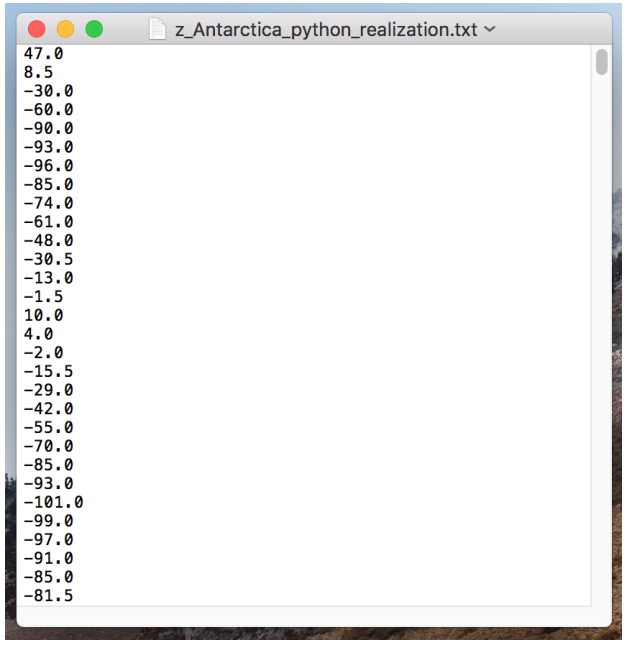
1. Prepare and read the prior materials
2. Run the tree construction program
3. Paste the files created by tree construction program into the simulation folder
4. Run the simulation program
5. Output realizations
6. Evaluate realizations

Example: Antarctica Topography Modeling (2D)

Example: Antarctica Topography Simulation Prepare Prior Materials

Our tree-based direct sampling (TDS) can read file in bmp and txt format.

The user should compile their own reading function in the training and simulation program.



2D topography model in txt format

In this txt file, each line stores a value at a certain pixel.

Consider the size of realization is 1,202×1,202, there are 1,202×1,202=1,444,802 lines in this file.

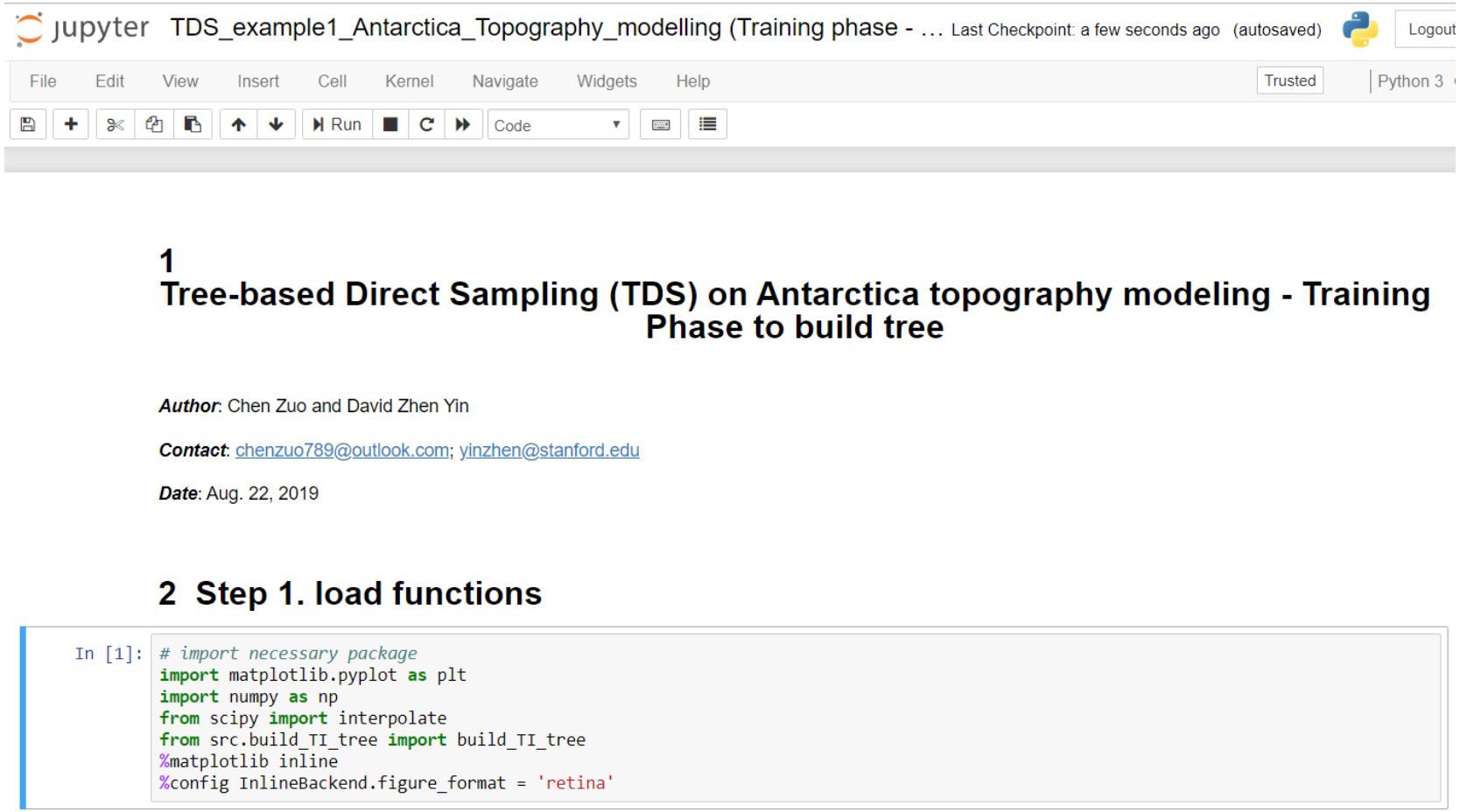
Step 1: Antarctica Topography Simulation Tree Construction Phase

Cell Index	Workflow	Tip
1	Import necessary package	
2	Set user-defined parameters	Three parameters are required to be specified.
3	Read training image	Users can write their own function
4	Record time stamp	
5	Find the processing area	We do not concern the ocean area in this case
6	Record time stamp	
7	Extract patterns via fixed template	
8	Perform PCA to reduce dimension	The preserved variance is 80% in the example
9	Print scatter plot of patterns	Comment this cell during time recording
10	Get the reduced pattern dataset	
11	Calculate the radius within the pattern set	The radius is used to normalize the radius within clusters
12	Build the cluster tree	The core cell of training program
13	Record time stamp	
14	Output file	
15	Record time stamp	

Step 1: Antarctica Topography Simulation Tree Construction Phase

Under directory "./Tree-based_DS/TDS_TrainingPhase", use Jupyter Notebook to open the *.ipynb file for TI tree construction.

TDS_example_Antarctica_Topography_modelling (Training phase - build tree).ipynb



Step 1: Antarctica Topography Simulation Tree Construction Phase

Following the Jupyter Notebook instructions step by step to set up input parameters for TI tree onstruction

In [3]:
```
# Read the training image
# these four parameter come from the user
Realization_Height = 1202    # the height of realization y-axid
Realization_Width = 1202     # the width of realization x-axix
knownUpbound = 3346          # the maximum value of elevation
knownBelowbound = -5000     # the minimum value of known elevation

Realization_row = np.loadtxt('z_Antarctica_python_realization.txt',skiprows=0)
# print('the size of realization is '+str(Realization_row.shape))
Realization_2D = Realization_row.reshape(Realization_Height,Realization_Width)
plt.imshow(Realization_2D,cmap = 'jet', vmin=-7000, vmax=3000)
plt.title('Antarctica bed elevation map')
plt.colorbar()
```

Step 1: Antarctica Topography Simulation
Tree Construction Phase

Following the Jupyter Notebook instructions to set up input parameters for TI tree onstruction

In [2]:
```
# artificially eliminate the occean area on the top right corner
# the area whose elevation below this value is not trained
ArtificalThreshold = -1000

# set the radius of template the size of template is 2*radius+1
template_radius_x = 8
template_radius_y = 8

TreeHeight_max = 9
```

In [3]:
```
# Read the training image
# these four parameter come from the user
Realization_Height = 1202    # the height of realization y-axid
Realization_Width = 1202     # the width of realization x-axix
knownUpbound = 3346          # the maximum value of elevation
knownBelowbound = -5000     # the minimum value of known elevation

Realization_row = np.loadtxt('z_Antarctica_python_realization.txt',skiprows=0)
```

Set parameters

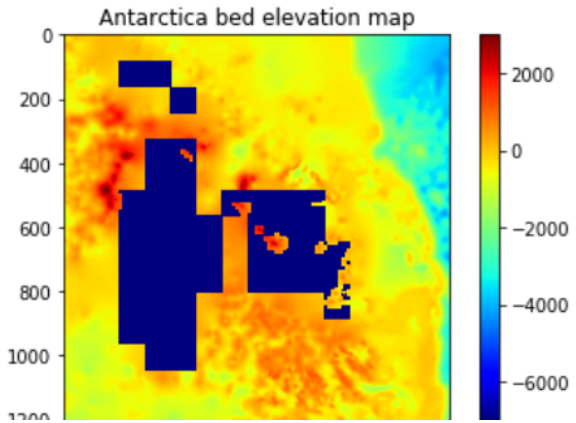
1) The actual size of template is *2*radius+1*
2) The height *h* divides patterns into 2^h clusters
3) We do not concern the ocean area in this case

Step 1: Antarctica Topography Simulation Tree Construction Phase

Following the Jupyter Notebook instruction to set up input parameters for TI tree onstruction

In [3]:
```
# Read the training image
# these four parameter come from the user
Realization_Height = 1202    # the height of realization y-axid
Realization_Width = 1202     # the width of realization x-axix
knownUpbound = 3346          # the maximum value of elevation
knownBelowbound = -5000     # the minimum value of known elevation

Realization_row = np.loadtxt('z_Antarctica_python_realization.txt',skiprows=0)
# print('the size of realization is '+str(Realization_row.shape))
Realization_2D = Realization_row.reshape(Realization_Height,Realization_Width)
plt.imshow(Realization_2D,cmap = 'jet', vmin=-7000, vmax=3000)
plt.title('Antarctica bed elevation map')
plt.colorbar()
plt.show()
```



This cell is applied to read a txt file.

These values come from background knowledge.

Users can write their own function to read certain training images.

Step 1: Antarctica Topography Simulation
Tree Construction Phase

Once all the input parameters are set, simply run Step 3 command to construct the tree using TI.

4 Step 3. runing Training Phase to construct tree using training image

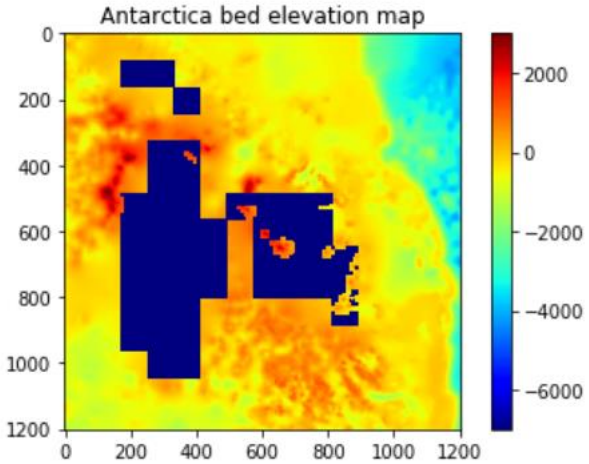
In [5]:
```
build_TI_tree(project_name, TI, TI_row, Realization_Height, Realization_Width, ArtificalThreshold, \
              template_radius_x, template_radius_y, TreeHeight_max, knownUpbound, knownBelowbound)
```

```
mission begins !
2019-09-19 20:39:55
The amount of pattern is 881351
The size of pattern is 289
running PCA
The reduced dimension is 1
```

Step 1: Antarctica Topography Simulation Tree Construction Phase

Input:

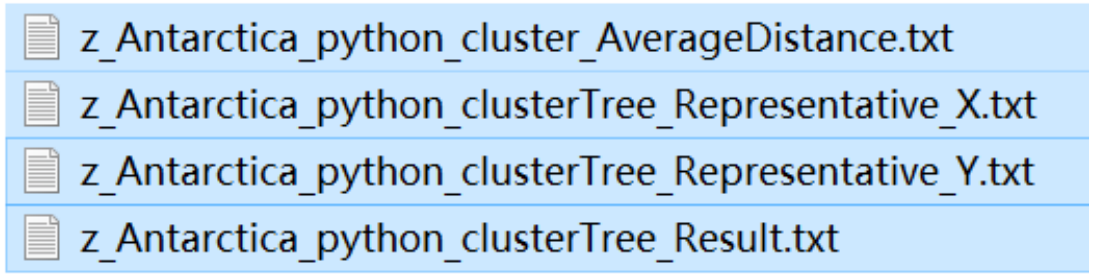
1) Training image



The training image is expressed as txt file in this example

Output:

1) z_Antarctica_python_cluster_AverageDistance.txt
2) z_Antarctica_python_clusterTree_Representative_X.txt
3) z_Antarctica_python_clusterTree_Representative_Y.txt
4) z_Antarctica_python_clusterTree_Result.txt



These four files are used to build the clustering tree.

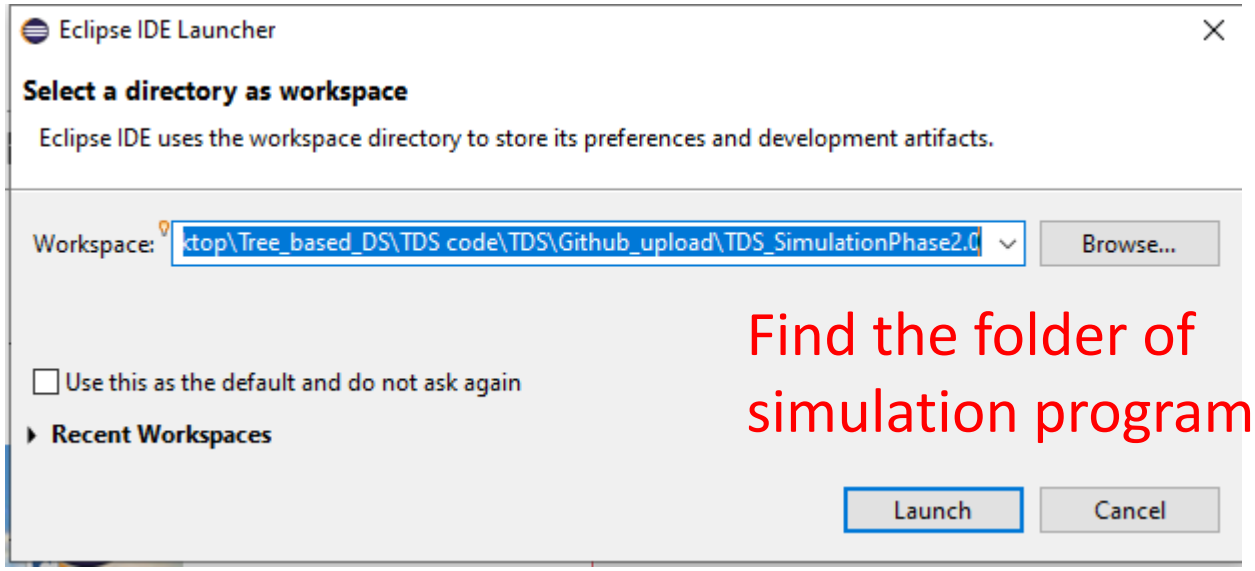
User-defined parameters:

1) Template size
2) The height of tree
3) The land elevation threshold

Step 2. Simulation Phase

Opening the Eclipse Java IDE, open the Simulation Phase under directory:
“./Tree-based_DS/TDS_SimulationPhase2.0”.

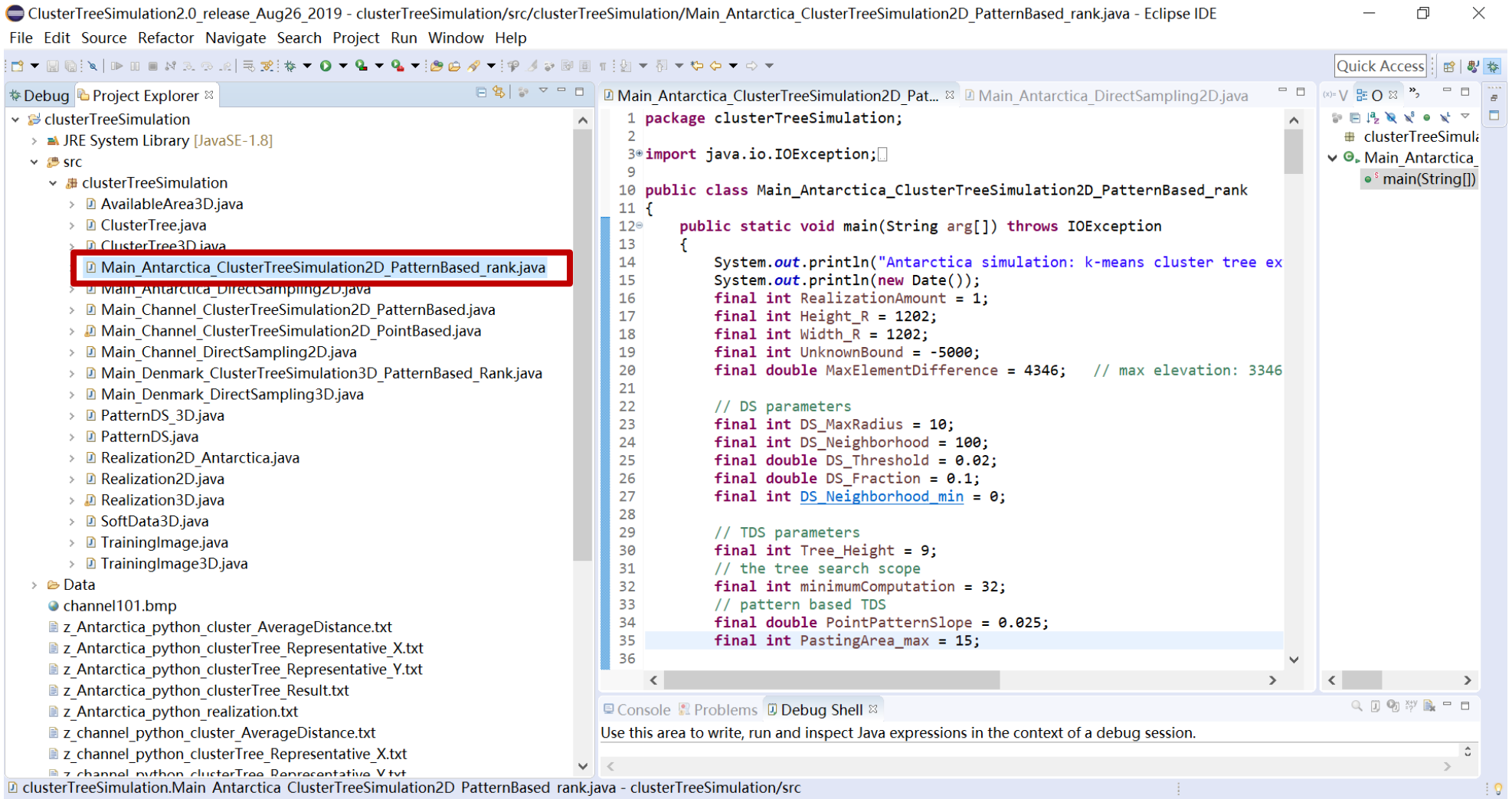
Icon of Eclipse Java



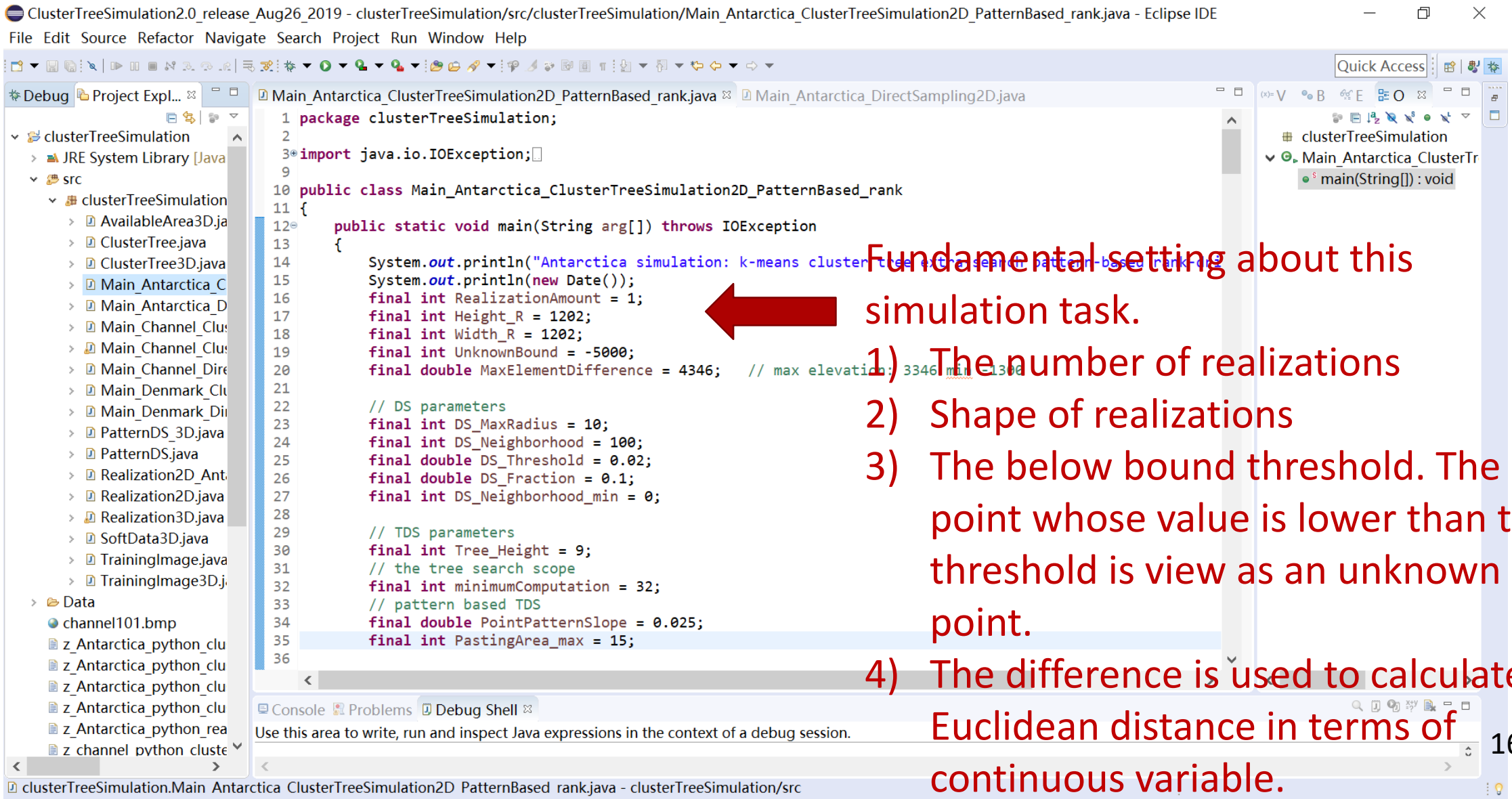
Launcher interface of Eclipse Java

Step 2: Antarctica Topography Simulation Simulation Phase

The TDS simulation program is **Main_Antarctica_ClusterTreeSimulation2D_PatternBased_rank.java**.
This function is used to generate 2D Antarctica topography realizations using TDS



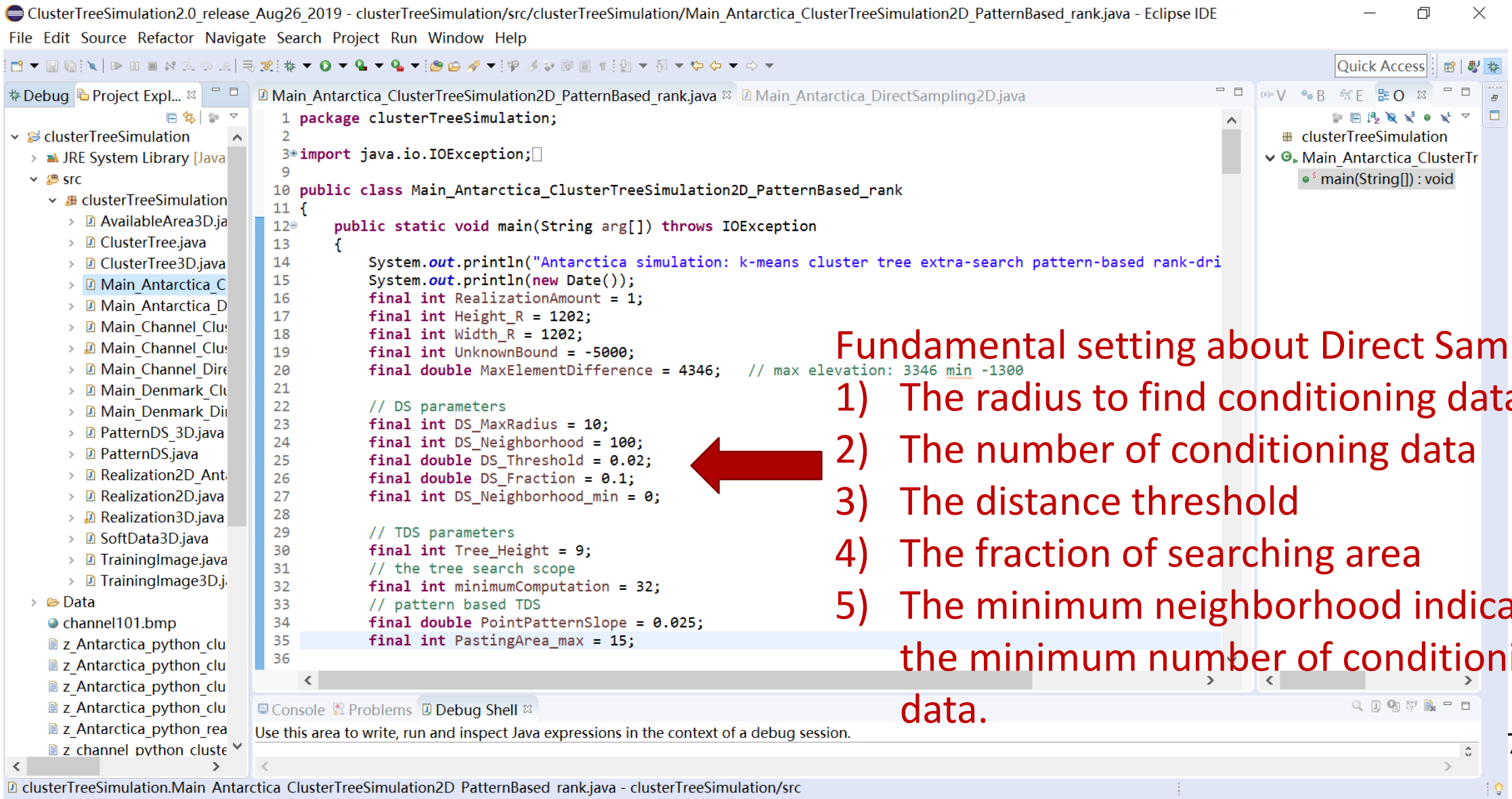
Step 2: Antarctica Topography Simulation Simulation Phase



Fundamental setting about this simulation task.

1) The number of realizations
2) Shape of realizations
3) The below bound threshold. The point whose value is lower than this threshold is view as an unknown point.
4) The difference is used to calculate the Euclidean distance in terms of continuous variable.

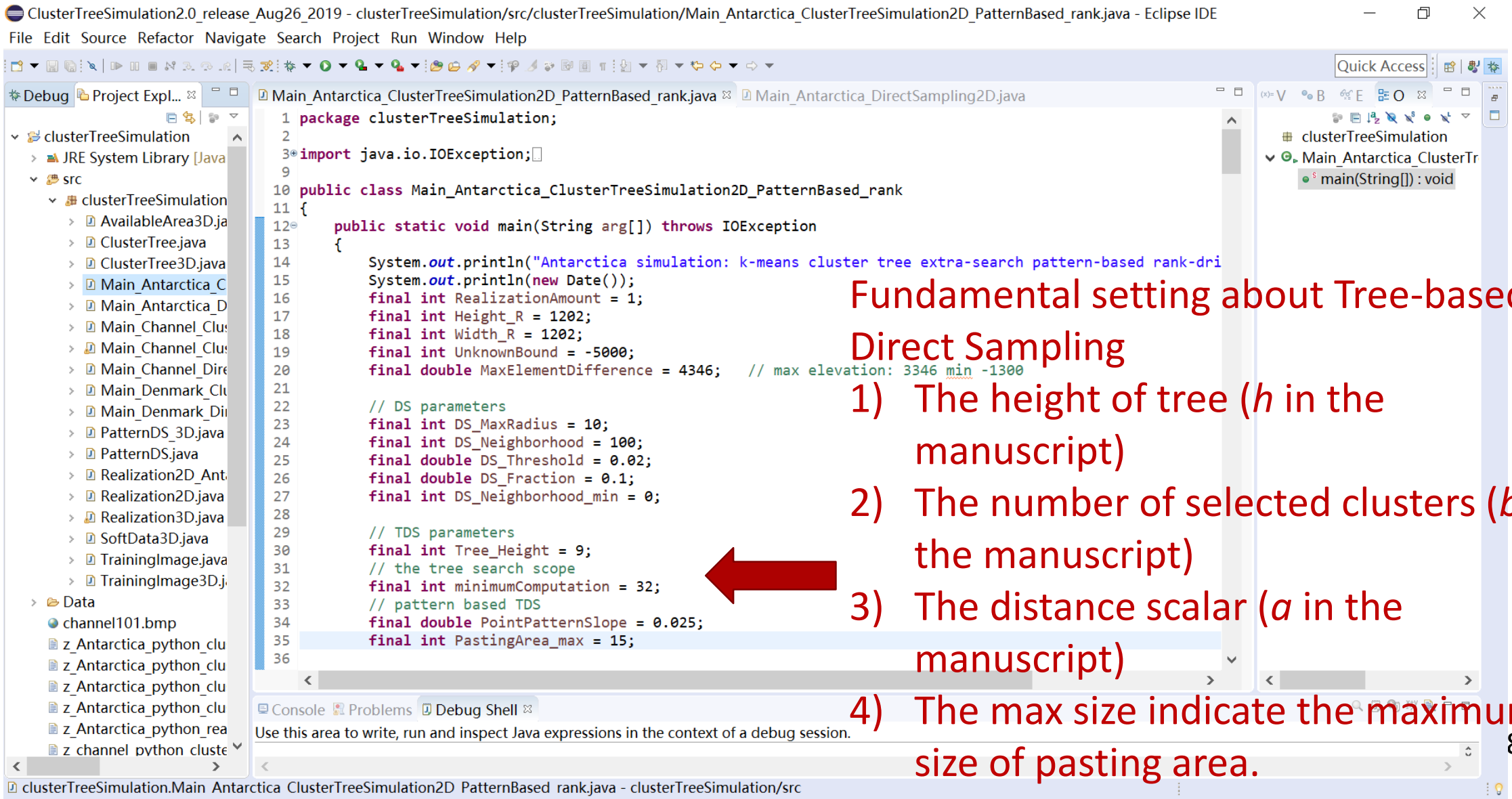
Step 2: Antarctica Topography Simulation Simulation Phase



Fundamental setting about Direct Sampling

1) The radius to find conditioning data
2) The number of conditioning data
3) The distance threshold
4) The fraction of searching area
5) The minimum neighborhood indicate the minimum number of conditioning data.

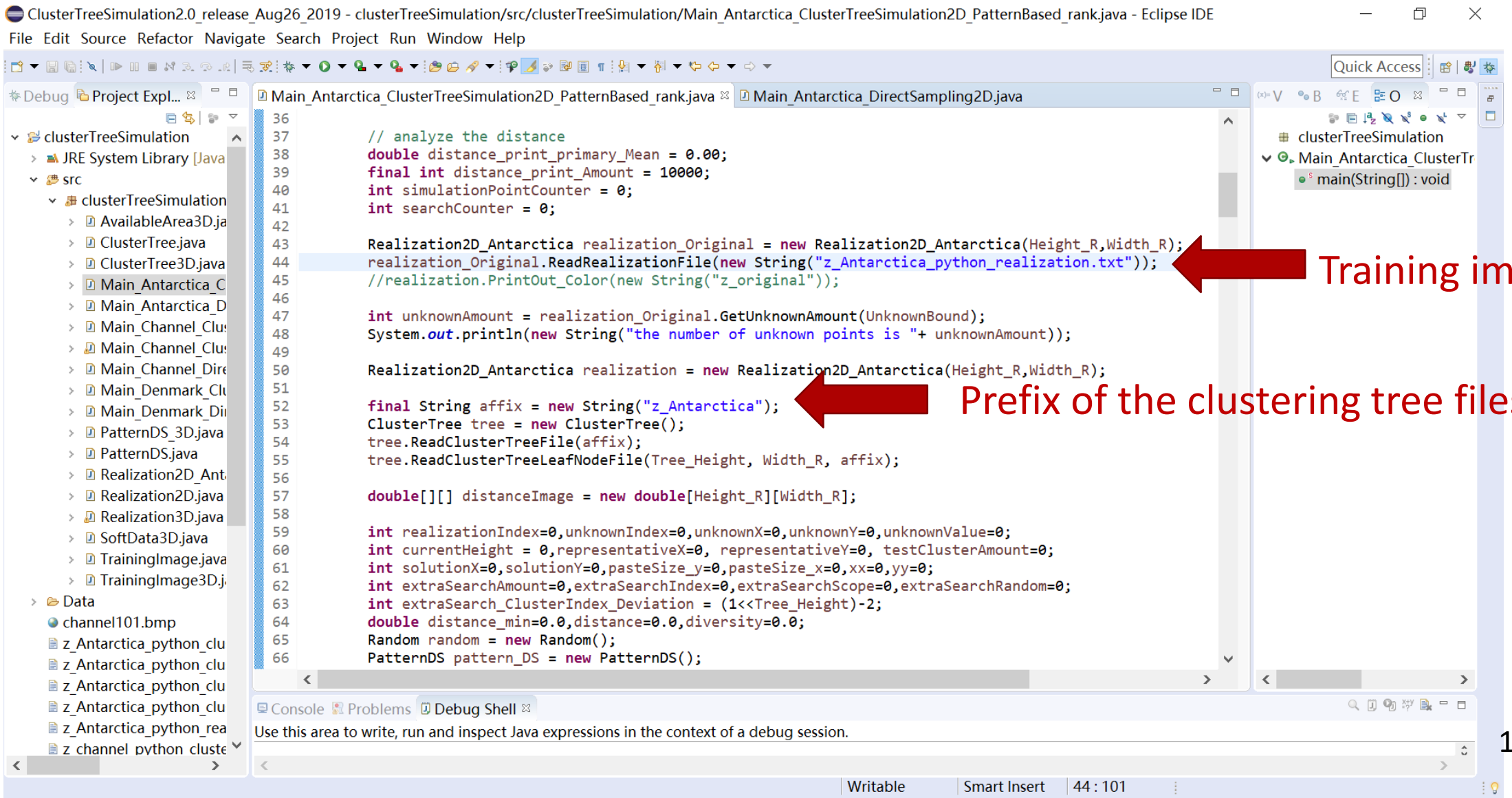
Step 2: Antarctica Topography Simulation Simulation Phase



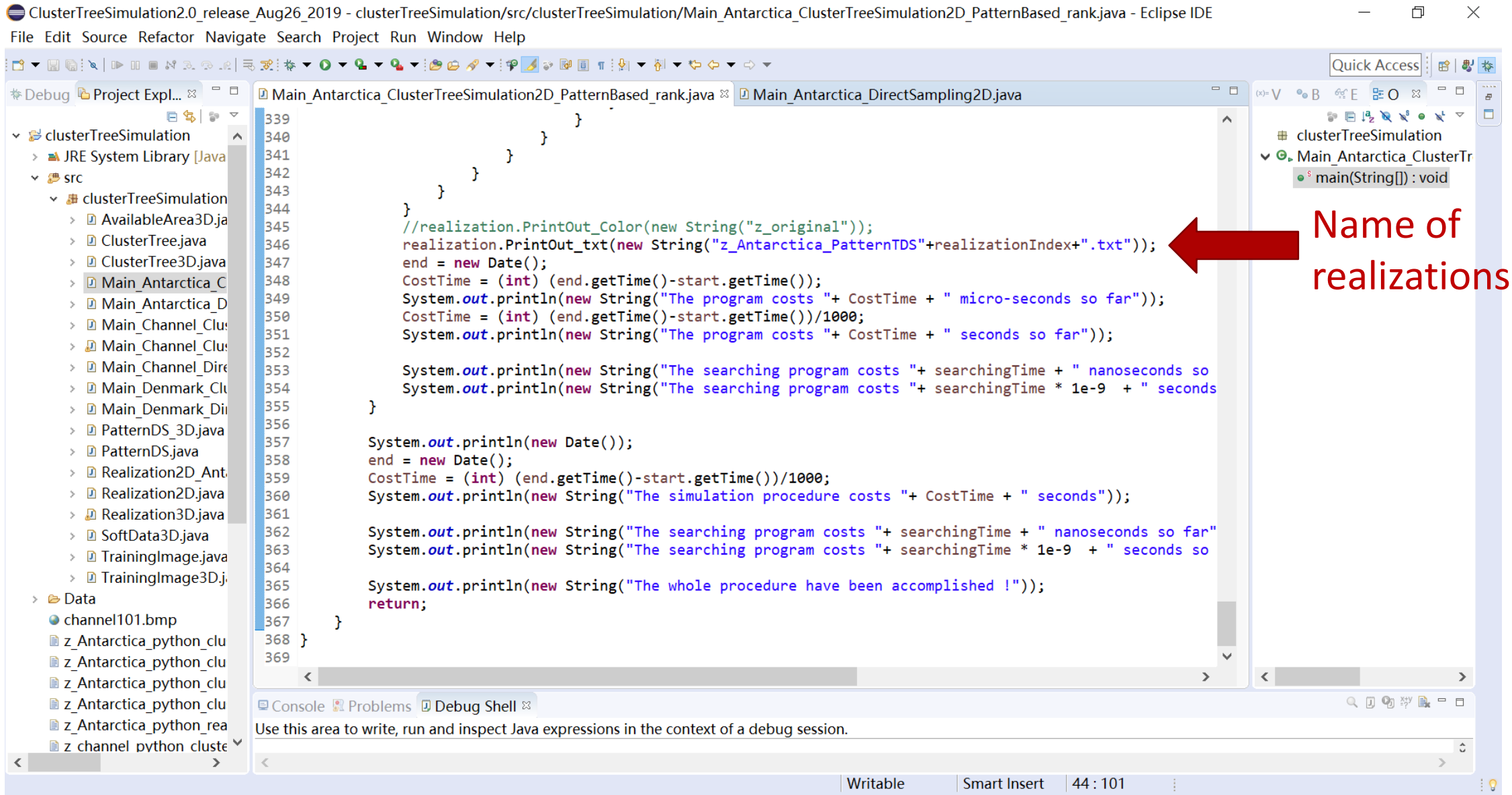
Fundamental setting about Tree-based Direct Sampling

1) The height of tree (*h* in the manuscript)
2) The number of selected clusters (*b* in the manuscript)
3) The distance scalar (*a* in the manuscript)
4) The max size indicate the maximum size of pasting area.

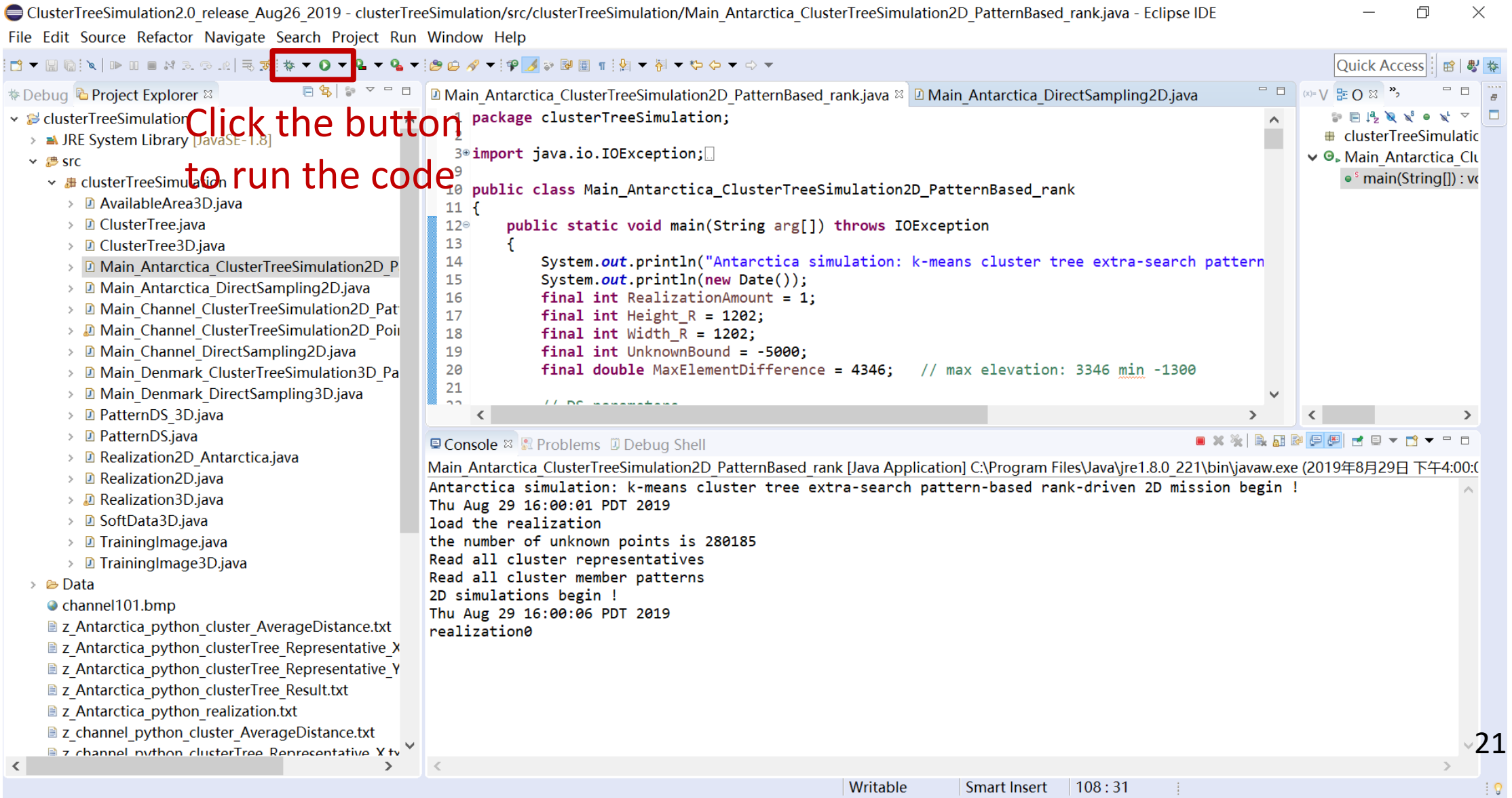
Step 2: Antarctica Topography Simulation Simulation Phase



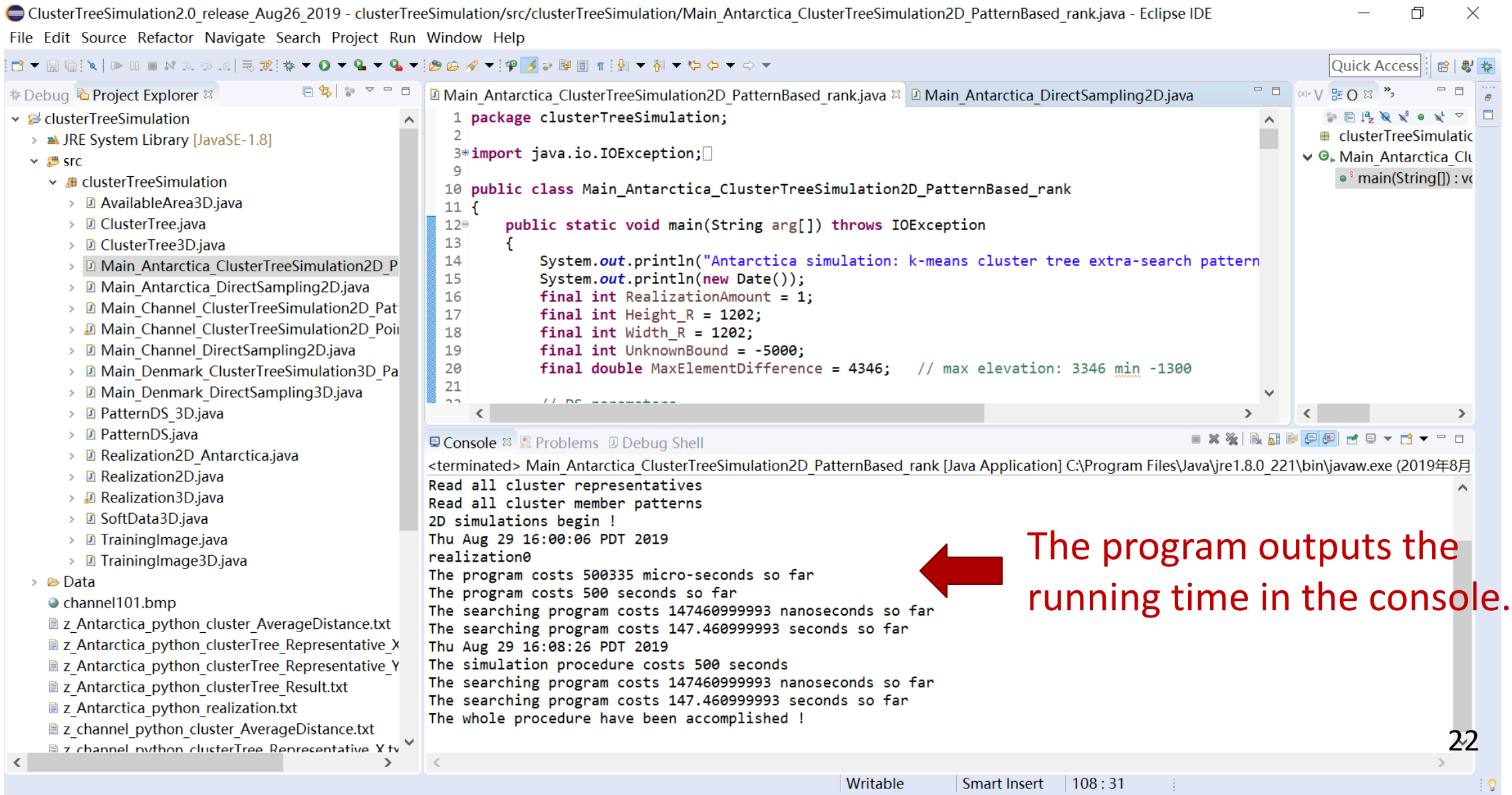
Step 2: Antarctica Topography Simulation Simulation Phase



Step 2: Antarctica Topography Simulation Simulation Phase



Step 2: Antarctica Topography Simulation Simulation Phase



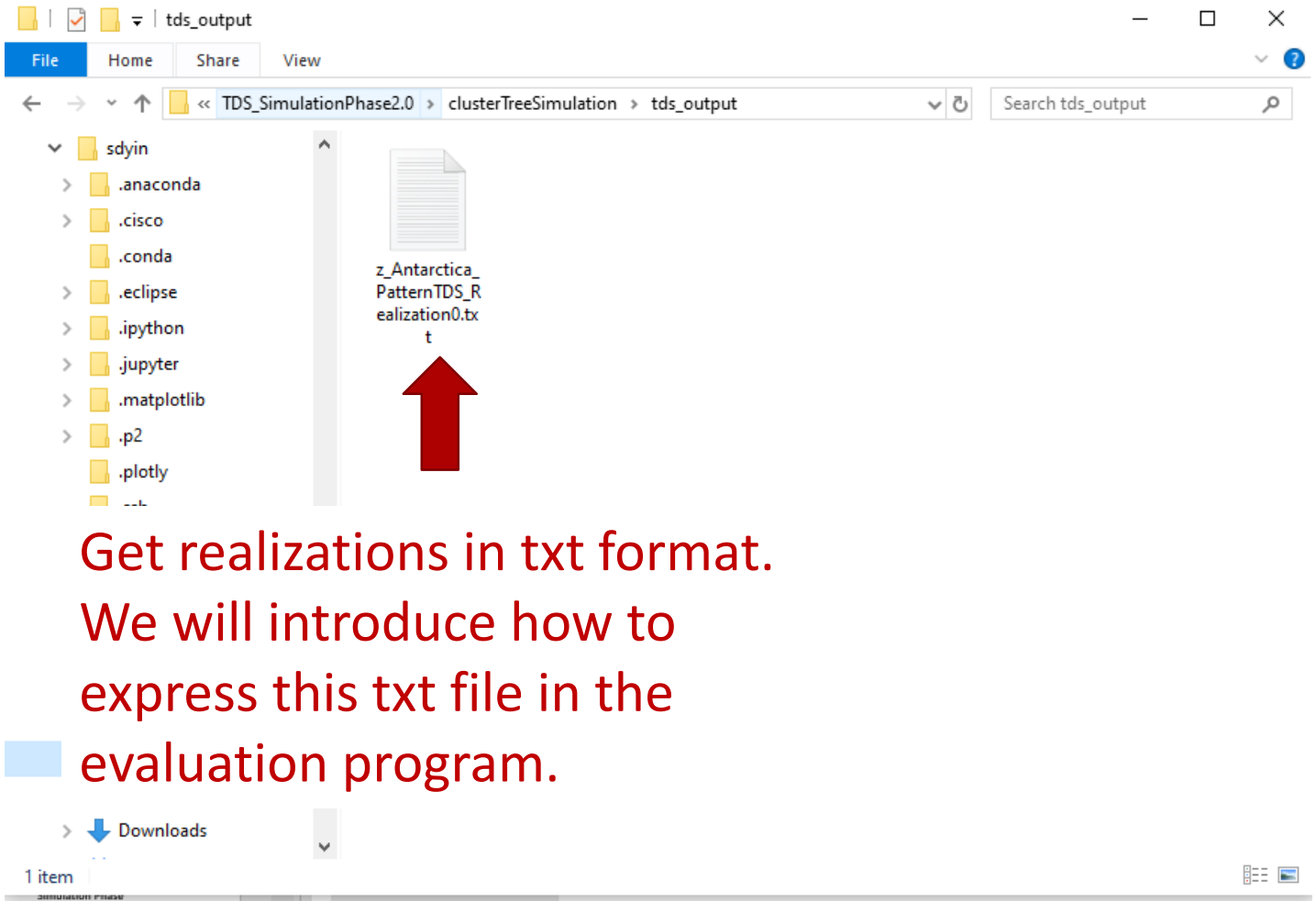
The program outputs the running time in the console.

Step 2: Antarctica Topography Simulation Simulation Phase

Results:

The TDS simulated realizations will be stored at directory:

"/Tree-based_DS/TDS_SimulationPhase2.0/clusterTreeSimulation/tds_output"



Get realizations in txt format.
We will introduce how to express this txt file in the evaluation program.